

What you need to know when selecting a chart

By Daniel Jebaraj, Vice President
SynCFusion, Inc.

Turn the pages of any magazine for developers and you will see ads for charts. Page after page is filled with images that look almost too good to be true. More than a few appear touched up with Photoshop to provide that extra glint. In this article, we will look at what makes a good chart beyond the obvious look and feel aspects.

Types of charts available out of the box

There are over 30 chart types that are used commonly in business applications (excluding specialized charts for financial and other applications we will address further in this paper). Some of the most common types include bar, column, area, line, scatter, bubble, pie, pyramid, funnel, polar, radar, Gantt, and tornado. Additionally there are stacked variants for many of these charts—stacked area, stacked bar, etc.

You will save a lot of time if the charts that you are most interested in are available out of the box. Also, be sure to test combinations of the different chart types. Chart types that can be plotted on similar axes should be supported together.

Axis properties

There are a few things you should be able to do with a chart's axis: You should be able to reverse its direction, shift its origin, include major and minor tick marks, customize individual tick marks, and apply labels to points.

Charts should also support automatic range calculation. Bonus points if the range calculation is friendly—calculated ranges should be readable.

Another important feature is the concept of a scale break. This is useful in cases where a section of data is not to be represented on the chart. Breaks are typically represented with a visual indicator on the chart.

Support for multiple chart areas

There is often a need to display multiple charts side-by-side. While it is possible to achieve this with multiple chart instances, it is not usually the best option. Lining up multiple charts can be quite tricky. Range calculations may not be identical. Label positions and spacing can be slightly different. It is much easier if the chart natively supports the concept of multiple chart areas.

Chart areas are essentially charts within charts. Multiple layout options should be available for positioning chart areas relative to each other. The chart should also offer options to synchronize multiple chart areas so they line up perfectly. This will allow related data to be plotted with a shared axis.

Support for data binding

When selecting a chart for frameworks such as WPF or Silverlight, it is important to ensure that the chart has solid support for data binding that is in-line with expectations on the framework of choice. It is also important to check if there is built-in understanding of notification interfaces such as `INotifyPropertyChanged` and `INotifyCollectionChanged`. Support for these interfaces can greatly reduce the amount of code you have to write to enable common features. It is also important to test for the correct handling of empty values when binding.

Drill-down display

The chart should support drilling down. Drilling down involves the provision of out-of-the-box options such as replacing data in the chart with data from a new context. For instance, clicking on data for a state loads detailed information for that state. Some charts may additionally offer an in-line option such as a rich Tooltip. It should also be possible to obtain the correct context and then display your own user interface for the drill-down.

Performance

Performance aspects to consider include load time, total memory usage, time taken to add a point to a series (to be tested when the series contains a high number of points already), time taken to remove a point from a series, time taken to add a series, time taken to remove a series, and time taken to add or remove chart areas.

It is also important to pay careful attention to the optimization being performed by the chart in order to display large numbers of data points. In some cases, charts will simply compress this data and then display compressed data with great performance characteristics but with a resulting loss in fidelity. This may or may not be important, depending on your needs.

As a general rule, test with at least three to four times the amount of data you expect to be working with.

3-D charting

Three-dimensional charts can be broadly classified as two-dimensional charts displayed with 3-D visuals, but true 3-D charts are plotted with z-axis values. If you are interested in true 3-D charting, be sure to test the functionality. Most business-oriented charting packages do not focus on true 3-D charts; they may offer a few 3-D charts, but most advertised 3-D features are related to the 3-D visualization of 2-D data.

Financial charting

Several chart types are specific to the financial industry; though some of them may be useful in other domains as well. These include candle, high-low, high-low-open-close, Kagi, Renko, box-and-whisker, point-and-figure, and three line break. In addition to these types, you may also be interested in built-in support for technical indicators such as Bollinger bands, MACD, and simple and exponential averages. Many high-end charting packages offer support for these and other technical indicators.

Statistical charting

If your work involves statistical analysis, built-in support for such analysis will come in handy. High-end charting packages typically provide support for calculations such as mean, median, standard deviation, variance, correlation coefficient, and covariance. Packages may also provide support for analysis of variance and displaying normal distributions.

Native support for dates

One of the most important and often absent features is native support for dates. Many charts on the market treat dates as doubles. This makes it very difficult to work with dates. If dates are natively supported, you should be able to specify axis intervals in terms of date intervals such as months and weeks. It will also be possible to easily format dates for display. This is a key feature that any chart you consider should have.

Annotations

Charts should provide for the display of annotations on any chart point. It should be possible to position annotations relative to data points or at an absolute position. A variety of different shapes and labels should be supported out of the box, and you should be able to provide your own shapes and labels.

Legends

Chart legends are important elements of charts. They greatly enhance their visual appeal and readability. You should be able to position legends at any location relative to a chart such as the bottom, top, left, or right. Some charts also allow for free positioning or floating.

All visual aspects of a legend need to be customizable. It should be possible to control the display of a series by using a check box or similar element associated with a legend item.

User interaction

When users interact with charts, they typically expect to see rich ToolTip information. Multiple ToolTip options should be supported out of the box. There should also be support for custom ToolTips.

Charts should also allow users to zoom in and easily control the zoom level. You should test zooming-related features carefully to see if the axes scale as expected when you zoom.

You should be able to interactively highlight a section of a chart. The selected range should be available for easy access from other code, including data-binding code. Users also need to move points and series when enabled. This allows you to enable users to interactively modify data, if this is a use-case specific to your needs.

Export

It should be possible to export the chart to common image formats such as PNG, BMP, GIF, and JPEG. High-end packages also allow for exporting charts to Microsoft Excel format. When exporting to Excel, be sure to test if the export is simple HTML (not a desirable feature since Excel will warn users that the file is not in the expected format) or a compliant and technically valid Excel file. Many packages that advertise Excel support provide simple HTML exporting, thereby providing a less than optimal user experience.

Serialization

There should be built-in support for serializing all chart setting data to XML or binary formats. There should be built-in support for versioning that will allow you to open older settings in newer versions of the chart control.

Pre-built configuration dialogs

Pre-built dialogs are similar to the chart settings dialog in Microsoft Excel. They allow end users to configure common chart settings. They take quite a bit of work to build and require you to understand the internals of the chart package. It is a big win if they are built-in.

Excel-like sparklines

Sparklines are tiny charts that display trends in Excel cells. Excel allows for thousands of sparklines to be displayed. Some charts support a sparkline display mode. Key aspects to look for are fidelity with the Excel look and feel and the ability to display hundreds if not thousands of these charts. Since sparklines are used to indicate micro-trends, they have to be resource friendly.

Localization

All images and text displayed by a chart should be easy to localize. Bonus points if there is built-in support for your commonly used locales.

Right-to-left

It is usually possible to tweak axis settings to achieve a right-to-left (RTL) display. It is a plus if this support is built into the chart since there are always edge cases that you will have to deal with when limited to working with axis positioning only. If RTL support is baked-in, then aspects such as adjustment for proper RTL label display will also be built-in, saving you extra effort.

Viewport filters

One of the most popular charts in recent times is the Google finance chart. It combines two charts with one acting as a viewport filter. In the case of the Google finance chart, one chart serves as a date range filter. The other chart displays detailed information for the range specified by the viewport filter. This chart is a very friendly way to visualize a lot of data. If your users have not asked for it, they will ask for it at some point soon. This feature is possible to emulate with multiple charts if it is not built-in, but having it built-in will save a lot of time and effort.

Printing

There should be built-in support for printing and print preview. In addition, look for the provision of special color palettes for printing. For instance, if you will be printing in grayscale most of the time, it will greatly help to have support for a grayscale palette that you can switch to when printing.

Download a 30-day free evaluation of
Essential Studio Enterprise Edition at:

www.syncfusion.com/evaluation

